sparkfun$_{q}wiic_{r}elay_{p}y$

Release 0.0.1

Jul 14, 2021

Contents:

1 Contents **3**

2 Supported Platforms **5**

3 Dependencies **7**

4 Documentation **9**

5 Installation **11**
- 5.1 PyPi Installation . . . 11
- 5.2 Local Installation . . . 11

6 Example Use **13**

7 Table of Contents **15**
- 7.1 API Reference . . . 15
  - 7.1.1 qwiic_relay . . . 15
- 7.2 Example 1 . . . 17
- 7.3 Example 2 . . . 19

8 Indices and tables **21**

Python Module Index **23**

Index **25**

Python module for the Qwiic Relays, Listed below

- SparkFun Qwiic Single Relay
- SparkFun Qwiic Quad Relay
- SparkFun Qwiic Quad Solid State Relay
- SparkFun Qwiic Dual Solid State Relay

This package can be used in conjunction with the overall SparkFun qwiic Python Package

New to qwiic? Take a look at the entire SparkFun qwiic ecosystem.

CHAPTER 1

Contents

- *Supported Platforms*
- *Dependencies*
- *Installation*
- *Documentation*
- *Example Use*

CHAPTER 2

Supported Platforms

The Qwiic Relay Python package current supports the following platforms:

- Raspberry Pi

CHAPTER 3

Dependencies

This driver package depends on the qwiic I2C driver: Qwiic_I2C_Py

CHAPTER 4

Documentation

The SparkFun Qwiic Relay module documentation is hosted at ReadTheDocs

CHAPTER 5

Installation

5.1 PyPi Installation

This repository is hosted on PyPi as the sparkfun-qwiic-relay package. On systems that support PyPi installation via pip, this library is installed using the following commands

For all users (note: the user must have sudo privileges):

```
sudo pip install sparkfun-qwiic-relay
```

For the current user:

```
pip install sparkfun-qwiic-relay
```

5.2 Local Installation

To install, make sure the `setuptools` package is installed on the system.

Direct installation at the command line:

```
python setup.py install
```

To build a package for use with pip:

```
python setup.py sdist
```

A package file is built and placed in a subdirectory called dist. This package file can be installed using pip.

```
cd dist
pip install sparkfun_qwiic_relay-<version>.tar.gz
```

CHAPTER 6

Example Use

See the examples directory for more detailed use examples.

```
from __future__ import print_function
import qwiic_relay
import time
import sys

myRelays = qwiic_relay.QwiicRelay()

def runExample():

    print("\nSparkFun Qwiic Relay Example 1\n")

    if myRelays.begin() == False:
        print("The Qwiic Relay isn't connected to the system. Please check your␣
↪connection", \
            file=sys.stderr)
        return

    #Turn on relays one and three
    myRelays.set_relay_on(1)
    myRelays.set_relay_on(3)
    time.sleep(1)

    #Print the status of all relays
    for relayNum in range(4):
        current_status = None
        if myRelays.get_relay_state(relayNum) is True:
            current_status = "On"
        else:
            current_status = "Off"
        print("Status 1: " + current_status + "\n")

    #Turn off 1 and 3, turn on 2 and 4
```

(continues on next page)

(continued from previous page)

```
    myRelays.set_relay_off(1)
    myRelays.set_relay_on(2)
    myRelays.set_relay_off(3)
    myRelays.set_relay_on(4)
    time.sleep(1)


    #Turn all relays on, then turn them all off
    myRelays.set_all_relays_on()
    time.sleep(1)

    myRelays.set_all_relays_off()



if __name__ == '__main__':
    try:
        runExample()
    except (KeyboardInterrupt, SystemExit) as exErr:
        print("\nEnding Example 1")
        sys.exit(0)
```

CHAPTER 7

Table of Contents

7.1 API Reference

7.1.1 qwiic_relay

Python module for the SparkFun Qwiic Single Relay, SparkFun Qwiic Quad Relay, SparkFun Qwiic Dual Solid State Relay, SparkFun Qwiic Quad Solid State Relay

This package can be used in conjunction with the overall SparkFun qwiic Python Package

New to qwiic? Take a look at the entire SparkFun qwiic ecosystem.

class qwiic_relay.**QwiicRelay**(*address=None*, *i2c_driver=None*)

Parameters

- **address** – The I2C address to use for the device. If not provided, the default address is used.
- **i2c_driver** – An existing i2c driver object. If not provided a driver object is created.

Returns The Qwiic Relay device object.

Return type Object

begin()

Initialize the operation of the relay

Returns Returns true of the initialization was successful, otherwise False.

Return type bool

connected

Determine if the Qwiic Relay is connected to the system.

Returns True if the device is connected, otherwise False.

Return type bool

get_relay_state(*relayNum=None*)
Returns true if the relay is currently on, and false if it is off.

Returns Status of the relay

Return type bool

get_slow_pwm(*relayNum*)
Gets the value for the slow PWM signal. Can be anywhere from 0 (off) to 120 (on).

Param The relay to get the PWM signal of

Returns The value of the PWM signal, a value between 0 and 120

Return type bool

get_version()
Returns the firmware version for the Single Relay

Returns The firmware version

Return type string

is_connected()
Determine if the Qwiic Relay is connected to the system.

Returns True if the device is connected, otherwise False.

Return type bool

set_all_relays_off()
Turn's off all relays. This command does nothing for the single relay

Param The relay to turn off

Returns successful I2C transaction

Return type bool

set_all_relays_on()
Turn's on all relays. This command does nothing for the single relay

Param The relay to turn on

Returns successful I2C transaction

Return type bool

set_relay_off(*relayNum=None*)
Turn's off a relay,if we're using a single relay, do not pass in a relay number

Param The relay to turn off

Returns successful I2C transaction

Return type bool

set_relay_on(*relayNum=None*)
Turn's on a relay,if we're using a single relay, do not pass in a relay number

Param The relay to turn on

Returns successful I2C transaction

Return type bool

set_slow_pwm(*relayNum*, *pwmValue*)
Sets the value for the slow PWM signal. Can be anywhere from 0 (off) to 120 (on). A full cycle takes 1 second.

Param The relay to set the PWM signal of

Param The value of the PWM signal, a value between 0 and 120

Returns successful I2C transaction

Return type bool

version
Returns the firmware version for the Single Relay

Returns The firmware version

Return type string

7.2 Example 1

Listing 1: examples/qwiic_relay_ex1.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# qwiic_relay_ex1.py
#
# Example that shows the basics of using the quad and dual relays.
#-----------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, July 2020
#
# This python library supports the SparkFun Electronics qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2020 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```

(continues on next page)

(continued from previous page)

```
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#==============================================================================
# Example 1
#

from __future__ import print_function
import qwiic_relay
import time
import sys


QUAD_RELAY = 0x6D
SINGLE_RELAY = 0x18
QUAD_SOLID_STATE_RELAY = 0x08

#Be sure to initialize your relay with the proper address.
myRelays = qwiic_relay.QwiicRelay(QUAD_SOLID_STATE_RELAY)

def runExample():

    print("\nSparkFun Qwiic Relay Example 1\n")

    if myRelays.begin() == False:
        print("The Qwiic Relay isn't connected to the system. Please check your␣
↪connection", \
            file=sys.stderr)
        return

    #Turn on relays one and three
    myRelays.set_relay_on(1)
    myRelays.set_relay_on(3)
    time.sleep(1)

    #Print the status of all relays
    for relayNum in range(4):
        current_status = None
        if myRelays.get_relay_state(relayNum) is True:
            current_status = "On"
        else:
            current_status = "Off"
        print("Status 1: " + current_status + "\n")

    #Turn off 1 and 3, turn on 2 and 4
    myRelays.set_relay_off(1)
    myRelays.set_relay_on(2)
    myRelays.set_relay_off(3)
    myRelays.set_relay_on(4)
    time.sleep(1)


    #Turn all relays on, then turn them all off
    myRelays.set_all_relays_on()
    time.sleep(1)

    myRelays.set_all_relays_off()

```

(continues on next page)

(continued from previous page)

```
92
93
94  if __name__ == '__main__':
95      try:
96          runExample()
97      except (KeyboardInterrupt, SystemExit) as exErr:
98          print("\nEnding Example 1")
99          sys.exit(0)
```

7.3 Example 2

Listing 2: examples/qwiic_relay_ex2.py

```
1   #!/usr/bin/env python
2   #-----------------------------------------------------------------------------
3   # top_phat_button_ex2.py
4   #
5   # Example that shows how to set and get the slow PWM value
6   #-----------------------------------------------------------------------
7   #
8   # Written by  SparkFun Electronics, April 2020
9   #
10  # This python library supports the SparkFun Electronics qwiic
11  # qwiic sensor/board ecosystem on a Raspberry Pi (and compatible) single
12  # board computers.
13  #
14  # More information on qwiic is at https://www.sparkfun.com/qwiic
15  #
16  # Do you like this library? Help support SparkFun. Buy a board!
17  #
18  #==================================================================================
19  # Copyright (c) 2019 SparkFun Electronics
20  #
21  # Permission is hereby granted, free of charge, to any person obtaining a copy
22  # of this software and associated documentation files (the "Software"), to deal
23  # in the Software without restriction, including without limitation the rights
24  # to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
25  # copies of the Software, and to permit persons to whom the Software is
26  # furnished to do so, subject to the following conditions:
27  #
28  # The above copyright notice and this permission notice shall be included in all
29  # copies or substantial portions of the Software.
30  #
31  # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
32  # IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
33  # FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
34  # AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
35  # LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
36  # OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
37  # SOFTWARE.
38  #==================================================================================
39  # Example 2
40  #
41
```

(continues on next page)

(continued from previous page)

```
from __future__ import print_function
import qwiic_relay
import time
import sys

QUAD_RELAY = 0x6D
SINGLE_RELAY = 0x18
QUAD_SOLID_STATE_RELAY = 0x08

#Be sure to initialize your relay with the proper address.
myRelays = qwiic_relay.QwiicRelay(QUAD_SOLID_STATE_RELAY)

def runExample():

    print("\nSparkFun Qwiic Relay Example 2\n")

    if myRelays.begin() == False:
        print("The Qwiic Relay isn't connected to the system. Please check your connection", \
            file=sys.stderr)
        return

    #Note that our range is 0-120 for setting a PWM value as there are only 120 times where the zero crossing relay can switch in one second

    myRelays.set_slow_pwm(1, 30) #25% duty cycle
    myRelays.set_slow_pwm(2, 60) #50% duty cycle
    myRelays.set_slow_pwm(3, 90) #75% duty cycle
    myRelays.set_slow_pwm(4, 120) #100% duty cycle

    #Print out our PWM values
    for relayNum in range(1, 5):
        pwmValue = myRelays.get_slow_pwm(relayNum)
        print("PWM Value for relay " + str(relayNum) + ": " + str(pwmValue))
    #Let the slow PWM run for a while
    time.sleep(15)


    #Set all relays off
    myRelays.set_slow_pwm(1, 0)
    myRelays.set_slow_pwm(2, 0)
    myRelays.set_slow_pwm(3, 0)
    myRelays.set_slow_pwm(4, 0)

if __name__ == '__main__':
    try:
        runExample()
    except (KeyboardInterrupt, SystemExit) as exErr:
        print("\nEnding Example 1")
        sys.exit(0)
```

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

q

qwiic_relay, 15

Index

B

begin() (*qwiic_relay.QwiicRelay method*), 15

C

connected (*qwiic_relay.QwiicRelay attribute*), 15

G

get_relay_state() (*qwiic_relay.QwiicRelay method*), 15
get_slow_pwm() (*qwiic_relay.QwiicRelay method*), 16
get_version() (*qwiic_relay.QwiicRelay method*), 16

I

is_connected() (*qwiic_relay.QwiicRelay method*), 16

Q

qwiic_relay (*module*), 15
QwiicRelay (*class in qwiic_relay*), 15

S

set_all_relays_off() (*qwiic_relay.QwiicRelay method*), 16
set_all_relays_on() (*qwiic_relay.QwiicRelay method*), 16
set_relay_off() (*qwiic_relay.QwiicRelay method*), 16
set_relay_on() (*qwiic_relay.QwiicRelay method*), 16
set_slow_pwm() (*qwiic_relay.QwiicRelay method*), 16

V

version (*qwiic_relay.QwiicRelay attribute*), 17